MTG-Link: filling gaps in draft genome assemblies with linked read data

Anne Guichard[1,2], Fabrice Legeai[1,2], Arthur Le Bars[2,3], Paul Yann Jay[4], Mathieu Joron[4], Denis Tagu[1] and Claire Lemaitre[2]

[1] INRAE, Agrocampus Ouest, Université de Rennes, IGEPP, F-35650 Le Rheu, France

[2] Univ Rennes, CNRS, Inria, IRISA - UMR 6074, F-35000 Rennes, France

[3] Station Biologique de Roscoff, Plateforme ABiMS, CNRS FR2424, F-29682, Roscoff, France

[4] CEFE, CNRS, Univ Montpellier, Univ Paul Valéry Montpellier 3, EPHE, IRD, F-34293 Montpellier, France

Corresponding author: anne.guichard@irisa.fr

Abstract

Current advancements of both second and third generation sequencing technologies contribute to the improvement of the assembly of most genomes. However, complete and accurate reconstruction of large non-model organism genomes remains challenging. In particular, the scaffolding step orders and orients contigs but generates undefined sequences between them, called *gaps*. Linked read technologies, such as the 10X Genomics Chromium platform, have a great potential for filling the gaps in draft genomes as they provide long-range information while maintaining the power and accuracy of short-read sequencing [1][2]. With these technologies, reads that have been sequenced from the same long DNA molecule (around 30-50 Kb) can be identified thanks to a small barcode sequence. Several tools have been developed for gap-filling with short or long read data [3][4], but to our knowledge, none uses the long-range information of the linked read data.

Here, we present MTG-Link, a novel gap-filling tool dedicated to linked read data generated by 10X Genomics Chromium technology. MTG-Link is a Python pipeline combining the local assembly tool MindTheGap [5] and an efficient read subsampling based on the barcode information. For each gap, it extracts the linked reads whose barcode is observed in the gap flanking sequences, and assembles them into contigs by traversing their de Bruijn graph. MTG-Link automatically tests different parameter values for gap-filling, in both forward and reverse orientations, and produces for each, whenever it is possible, a sequence assembly. After automatic qualitative evaluation of the best sequence assembly, it returns a GFA file, containing the gap-filled sequences of each gap. In order to speed up the process, MTG-Link uses a trivial parallelization scheme by giving each gap to a separate thread. We validated our approach on a set of simulated gaps from real datasets with various genome complexities, and showed that the read subsampling step of MTG-Link enables to get better gap assemblies in less CPU time than using MindTheGap on its own. We then applied MTG-Link on several individual genomes of a mimetic butterfly (*Heliconius numata*), where it significantly improved the contiguity of a 1.3 Mb locus of biological interest.

MTG-Link is freely available at https://github.com/anne-gcd/MTG-Link.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 764840, and from the French ANR ANR-18-CE02-0019 Supergene grant. We acknowledge the GenOuest bioinformatics core facility (https://www.genouest.org) for providing the computing infrastructure.

References

[1] Ott A. et al. Linked read technology for assembling large complex and polyploid genomes. *BMC Genomics*, 19:651, 2018.

[2] Weisenfeld N. I., Kumar V., Shah P., Church D. M., and Jaffe D. B. Direct determination of diploid genome sequences. *Genome Research*, 27:757–767, 2017.

[3] Paulino D., Warren R. L., Vandervalk B. P., Raymond A., Jackman S. D., and Birol I. Sealer: a scalable gap-closing application for finishing draft genomes. *BMC Bioinformatics*, 16:230, 2015.

[4] Xu G.-C. et al. LR-Gapcloser: a tiling path-based gap closer that uses long reads to complete genome assembly. *GigaScience*, 8:1–14, 2018.

[5] Risk G., Gouin A., Chikhi R., and Lemaitre C. MindTheGap: integrated detection and assembly of short and long insertions. *Bioinformatics*, 30(24):3451–3457, 2014.